



Apprentissage de relations spatiales pour la reconnaissance d'expressions mathématiques manuscrites en-ligne

Ahmad-Montaser A.M. Awal, Harold Mouchère, Christian Viard-Gaudin

► To cite this version:

Ahmad-Montaser A.M. Awal, Harold Mouchère, Christian Viard-Gaudin. Apprentissage de relations spatiales pour la reconnaissance d'expressions mathématiques manuscrites en-ligne. Colloque International Francophone sur l'Ecrit et le Document (CIFED2010), Mar 2010, France. hal-00490999

HAL Id: hal-00490999

<https://hal.science/hal-00490999>

Submitted on 10 Jun 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Apprentissage de relations spatiales pour la reconnaissance d'expressions mathématiques manuscrites en-ligne

Ahmad-Montaser Awal, Harold Mouchère et Christian Viard-Gaudin

IRCCyN/IVC – UMR CNRS 6597

Ecole polytechnique de l'université de Nantes

Rue Christian Pauc – BP 50609 – 44306 Nantes CEDEX 3 – France

{ahmad-montaser.awal, harold.mouchere, Christian.Viard-Gaudin}@univ-nantes.fr

RÉSUMÉ. Nous proposons dans cet article une nouvelle méthode d'analyse syntaxique et structurelle pour un système de reconnaissance d'expressions mathématiques manuscrites en-ligne. Une grammaire probabiliste est mise en place pour regrouper les hypothèses de segmentation/reconnaissance proposées par un segmenteur 2D. Les probabilités associées à la grammaire sont calculées à partir d'informations structurelles entre les symboles. Ces informations structurelles utilisent l'évaluation d'une relation spatiale entre les éléments intervenants dans chaque règle. L'apprentissage du système se fait en deux phases, d'abord l'apprentissage global du classifieur sans tenir compte de la grammaire, puis l'apprentissage des relations spatiales intervenant dans la grammaire. Ce système est entraîné et testé sur une large base synthétisée d'expressions, puis testé sur une base d'expressions réelles complexes.

MOTS-CLÉS : *Écriture manuscrite, expressions mathématiques, relations spatiales, analyse syntaxique.*

ABSTRACT. We propose in this paper a new method for syntactic and structural analysis in a system for the recognition of online handwritten mathematical expressions. A probabilistic grammar is used to group the hypotheses of segmentation/recognition proposed by a 2D segmenter. Probabilities associated to the grammar are based on structural information of symbols. This structural information use the evaluation of spatial relations between symbols intervening in each rule of the grammar. The system is trained in two steps, first a global learning is done in order to train the symbol classifier, then the spatial relations are learnt. This system is trained and tested with a large base of synthetic expressions. The system is also tested with a set of real complex expressions.

KEYWORDS: *handwriting, mathematical expressions, spatial relations, syntactic analysis.*

1. Introduction

L'utilisation de notations mathématiques est indispensable dans la documentation scientifique. De plus, les expressions mathématiques représentent un outil universel de communication entre les scientifiques. Bien que ces expressions puissent être trop compliquées pour les non-spécialistes, elles décrivent parfaitement les problèmes dans la plupart des domaines. Pour comprendre l'importance des expressions mathématiques, nous avons extrait toutes celles qui se trouvent dans les pages web de « Wikipedia » français. Un ensemble de près de 77 000 expressions ont été retrouvées dans 7 000 pages web. Grâce au progrès technologique de ces dernières années, les documents papiers sont de moins en moins utilisés. Ainsi, la tendance dans la communauté scientifique à utiliser les actes numériques a considérablement augmenté. Ce progrès exige un développement en parallèle des outils de saisie qui facilitent l'intégration d'expressions mathématiques dans ces documents.

De nombreux outils existants permettent la saisie d'expressions mathématiques dans les documents numériques. Cependant, une certaine expertise est exigée pour l'utilisation de ces outils. Certains symboles et fonctions mathématiques sont décrits par des mots clés prédéfinis qu'il faut mémoriser, c'est le cas avec Latex et MathML. De plus, l'expression qui est naturellement bidimensionnelle doit être écrite en une seule dimension, rendant la tâche plus complexe. D'autres outils, tels que Math Type, dépendent d'un environnement visuel pour ajouter des symboles à l'aide de la souris. Ces dépendances de mots clés ou de la souris augmentent le temps requis pour entrer une expression.

Les nouvelles technologies, telles que les tablettes pc, le tableau blanc interactif, le stylo numérique, etc. représentent des outils intéressants pour rendre la saisie d'expressions mathématiques plus facile. Ces outils doivent être accompagnés de systèmes de reconnaissance capables de transformer l'écriture manuscrite en format numérique. Plusieurs recherches émergent dans ce domaine avec des résultats prometteurs. Mais la plupart de ces recherches abordent le problème de reconnaissance d'expressions mathématiques comme un ensemble de sous-tâches séquentiellement indépendantes. Ce découpage permet de simplifier le problème de reconnaissance et d'isoler des sous problèmes. En conséquence des erreurs pourront être héritées d'une étape à l'autre sans qu'elles puissent être corrigées au cours de la reconnaissance.

L'objectif de nos travaux est de proposer un système de reconnaissance d'expressions mathématiques manuscrites en-lignes. Nous considérons la reconnaissance comme une optimisation simultanée de la segmentation, de la reconnaissance et de l'analyse structurelle et syntaxique des expressions mathématiques. Notre contribution dans cet article est de proposer une méthode d'analyse structurelle et syntaxique basée sur l'apprentissage de relations spatiales entre les symboles de l'expression mathématique.

Dans la suite nous introduisons la problématique de la reconnaissance d'expressions mathématiques. Puis nous détaillons notre architecture et la méthode proposée. Nous terminons avec quelques expérimentations et leurs résultats.

2. La reconnaissance d'expressions mathématiques

La reconnaissance d'une expression mathématique se déroule en trois étapes principales (Chan et al., 2000) : la segmentation, la reconnaissance de symboles, et l'interprétation de l'expression. Une expression mathématique est un arrangement complexe de symboles mathématiques en deux dimensions. Certains éléments tels que les fractions, les indices et les exposants introduisent des changements de directions et/ou de taille dans l'agencement des symboles, ce qui rend les processus de segmentation et de reconnaissance plus compliqués. Nous définissons un trait en tant qu'un ensemble de points dessinés entre un posé et levé de stylo. Nous considérons que chaque symbole consiste en un ou plusieurs traits séquentiels ou non-séquentiels. Contrairement à la plupart des systèmes actuels, l'architecture proposée supporte les traits non séquentiels dans un même symbole, ce qui autorise l'entrelacement temporel des symboles.

2.1. La segmentation de l'expression mathématique

Une bonne segmentation de l'expression mathématique est le point clé d'une bonne reconnaissance et interprétation. La segmentation consiste à regrouper les traits appartenant au même symbole. (Faure et al., 1990) propose deux modules pour appliquer des projections successives sur les axes X et Y, et un arbre de relations spatiales afin de regrouper les traits. Une méthode similaire a été proposée dans (Ha et al., 1995) mais en utilisant les boîtes englobantes des traits. (Smithies et al., 1999) ont développé un algorithme simple de regroupement pour la segmentation en ligne. Le système consiste à utiliser la reconnaissance de caractères afin de déterminer le taux de confiance pour chaque regroupement de traits possibles. On sélectionne ensuite celui avec le plus fort taux de confiance sachant que le niveau de confiance pour un groupe correspond au niveau de confiance minimum de chaque caractère reconnu pour chaque groupe. En plus de la reconnaissance de symboles, l'analyse syntaxique peut participer également à décider la segmentation (Yamamoto et al. 2006) et (Rhee et al., 2008). Pour réduire la complexité de cette optimisation simultanée des algorithmes comme « best first search » (Rhee et al., 2008), ou CYK (Yamamoto et al. 2006) peuvent être utilisés.

Après avoir regroupé les traits, l'étape de reconnaissance vise à identifier chaque regroupement en leur donnant un label de symboles mathématiques.

2.2. La reconnaissance de symboles mathématiques

Le nombre de symboles mathématiques est énorme comparé à du texte normal. Plus de 220 symboles sont utilisés dans les expressions mathématiques les plus fréquentes. Certains de ces symboles peuvent jouer des rôles différents selon leurs positions et le contexte où ils se trouvent. Néanmoins, cette étape peut être considérée comme un problème classique de reconnaissance de formes. Quelques systèmes existants (Chan et al., 2000) ou (Belaid et al., 1984) utilisent des méthodes structurelles pour la classification des symboles. Plusieurs systèmes, dont celui de (Nakayama, 1993) et (Rhee et al., 2008), utilisent la méthode d'appariement de motifs qui consiste à comparer l'objet à déterminer avec l'ensemble des images des symboles présents dans la base de données. Il existe de nombreuses méthodes pour accélérer les comparaisons mais celles-ci restent assez lourdes. Les réseaux de neurones artificiels sont aussi utilisés dans quelques systèmes (Marinkewitsch, 1991) ou (Dimitriadis et al., 1995). Ils montrent expérimentalement que cette technique est meilleure en terme de mémoire et de temps de traitement que les algorithmes de mise en correspondance. On peut également obtenir simultanément une segmentation et une reconnaissance en utilisant des méthodes statistiques comme les Modèles de Markov Cachés (Lehmberg et al., 1996) et (Yamamoto et al. 2006).

2.3. L'interprétation de l'expression mathématique

L'interprétation d'une expression mathématique consiste à analyser la structure géométrique de l'expression et puis à appliquer une analyse syntaxique. Le but de cette analyse est de trouver l'arbre syntaxique de l'expression. La structure géométrique d'une expression mathématique est très souvent plus complexe que celle d'un texte. Ainsi, un texte s'écrit systématiquement de gauche à droite alors que les symboles mathématiques peuvent s'écrire dans presque toutes les directions, comme le montre la Figure 1.



Figure 1 Différence de directions d'écriture entre un simple texte et une expression mathématique

Les relations spatiales entre les symboles sont cruciales pour l'interprétation des expressions. En effet, même si tous les symboles ont correctement été reconnus, il reste le problème de l'interprétation de la structure en deux dimensions de l'expression. Des ambiguïtés apparaissent aussi dans cette phase d'analyse comme :

Apprentissage de relations spatiales

- le rôle sémantique des symboles : Quelques symboles peuvent avoir plusieurs sens suivant le contexte (barre horizontale, point...) .
- la position relative des symboles : Dans les expressions mathématiques, les relations spatiales entre symboles correspondent à des relations logiques. Ainsi, les structures comme les exposants, les indices, les multiplications implicites, etc, sont indiquées par le contexte global de l'expression.

(Chan et al., 2000) proposent une méthode basée sur une « definite clause grammar » (grammaire qui permet d'exprimer les relations grammaticales). La DCG est utilisée pour définir un jeu de règle de remplacement pour analyser les expressions. Mais cette approche n'est pas très efficace car elle implique un grand nombre de retours en arrière. Ils ont proposé d'améliorer l'efficacité du système en utilisant des « left factored » règles. Plus récemment, (Garain et al., 2004) propose une méthode se basant sur une grammaire hors contexte. Avec des informations géométriques de l'expression, il effectue l'analyse à l'aide de la grammaire. Initialement, la structure de l'expression est construite en divisant l'expression récursivement en bande horizontale et verticale jusqu'à ce qu'un niveau atomique soit atteint (un seul symbole). Finalement, on fusionne chaque élément atomique en suivant les règles de production correspondant aux relations spatiales, afin d'obtenir l'équivalent LaTeX. (Zanibbi et al., 2002) proposent pour leur part une technique basée sur la transformation d'arbre. Ainsi, une recherche récursive permet d'identifier les structures linéaires de l'expression et l'on construit un arbre « baseline structure tree » qui est ensuite soumis à une analyse lexicale pour produire l'arbre final.

(Fitzgerald et al., 2006) présentent une approche appelée « Fuzzy Shift-Reduce Parsing » (FSRP). Cette méthode est construite sur une traditionnelle analyse ascendante, donc assure une vérification syntaxique efficace. La logique floue est introduite pour faire face aux imprécisions inhérentes à l'écriture manuscrite. (Genoe et al., 2005) proposent une technique originale. L'arbre de l'expression est immédiatement mis à jour pour chaque nouveau trait de l'utilisateur. Ils utilisent la logique floue à chaque étape de la reconnaissance pour pallier aux imprécisions de l'écriture manuscrite. Ils proposent un nouvel algorithme, le « Fuzzy Online Structural Analysis algorithm » (FOSA).

Dans (Yamamoto et al. 2006), il est proposé une grammaire probabiliste et structurelle. Chaque règle de la grammaire est reliée à une relation logique (au-dessus, en-dessous, ...). Une probabilité est aussi associée à chaque règle. Donc, la reconnaissance de l'expression est transformée en une recherche des règles maximisant la probabilité d'obtenir l'expression résultat.

3. Le système de reconnaissance

Dans l'architecture proposée nous essayons de compenser les problèmes de chacune des étapes de reconnaissance. Nous considérons la reconnaissance d'expressions mathématiques comme une optimisation simultanée de la segmentation, de la reconnaissance de symboles, et de l'interprétation.

L'apprentissage du système et la reconnaissance d'expressions mathématiques se font en utilisant la même architecture globale de reconnaiseur d'expressions proposée dans (Awal et al., 2009), voir Figure 2.

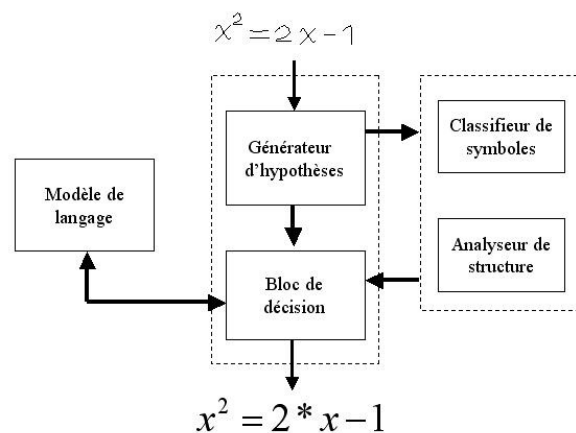


Figure 2 Architecture du système

Il consiste en :

- Un générateur d'hypothèses de symboles élaborant des combinaisons des traits. Un groupe de traits s'appelle une hypothèse de symbole (HS). Toutefois, beaucoup d'hypothèses de segmentation sont invalides. Elles correspondent soit à des sous-ensembles de symboles, lorsqu'un symbole est écrit en plusieurs traits, soit à des sur-ensemble de symboles lorsque le regroupement considéré de traits fusionne plusieurs symboles. Nous adoptons une extension d'un algorithme de programmation dynamique en deux dimensions permettant de gérer des regroupements de traits non-consécutifs.
- Un classifieur de symboles associant un score de reconnaissance, mais aussi une étiquette pour chacune des hypothèses. Comme nous avons proposé dans (Awal et al., 2009) nous utilisons un classifieur capable d'identifier les mauvaises hypothèses de symboles.
- Un analyseur de structure permettant d'obtenir des informations structurelles sur chaque hypothèse. Mais, comme nous l'avons vu, il existe des ambiguïtés, pour y pallier nous calculons des informations différentes selon le type de symboles. La

Figure 3 montre les différentes positions utilisées pour définir la ligne de base (y_0) et la hauteur (h) pour différents types de symboles.

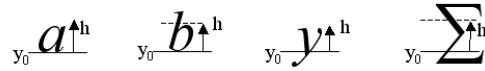


Figure 3 Lignes de base et hauteur de différents types de symboles

- Un modèle de langage défini par la grammaire de production d'expressions mathématiques. Chacune des règles de cette grammaire s'active si la relation associée a une probabilité forte par rapport aux autres règles.
- Un bloc de décision choisissant l'ensemble des hypothèses minimisant le coût global et respectant le modèle de langage. Le coût global est constitué des scores de reconnaissance des symboles et des scores d'activation obtenus de toutes les règles qui produisent l'expression finale.

3.1. Représentation de l'expression mathématique

Nous représentons une expression candidate par un arbre relationnel. Chaque nœud de cet arbre contient une hypothèse de regroupement de traits. La racine de cet arbre présente la solution proposée associée avec le coût global de cette solution, le type de relation entre ses nœuds fils, et son étiquette. De même façon, les nœuds non terminaux représentent des sous-expressions, leurs coûts, la relation qui regroupe ses nœud fils, et leurs étiquettes. Enfin, les nœuds terminaux sont les hypothèses de symboles proposés par le segmenteur 2D avec ces coûts de reconnaissance. Prenons un simple exemple, la grammaire suivante génère des arbres candidats similaires à ceux de la Figure 4. Les règles qui ne sont pas associées aux relations spatiales produisent les nœuds terminaux. Un type de relation est associé à chaque règle qui produit un nœud non-terminal :

Règle	Type de relation
$\text{sym} \leftarrow x, y, 1, 2, \dots$	
$\text{op} \leftarrow +, -, \times, \dots$	
$\text{formule} \leftarrow \text{subExp op sym}$	(operator)
$\text{subExp} \leftarrow \text{subExp op sym}$	(operator)
$\text{subExp} \leftarrow \text{sym sym}$	(superscript)
$\text{subExp} \leftarrow \text{sym}$	

Tableau 1 Exemple d'une grammaire simple

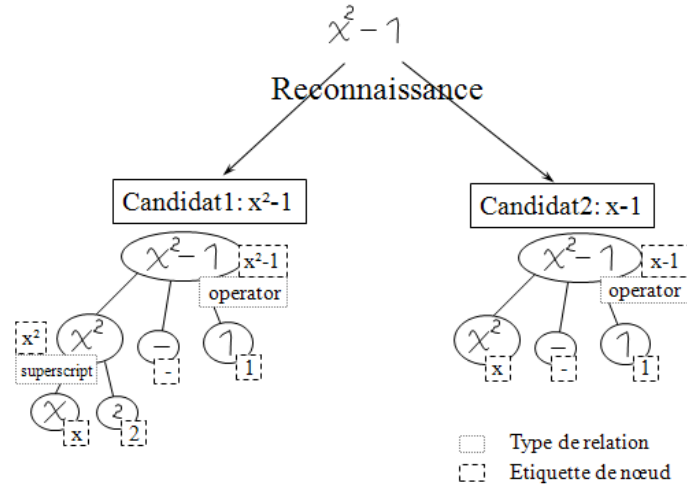


Figure 4 L'arbre relationnel de deux expressions candidates

Nous avons choisi d'introduire une modélisation spatiale des relations entre les composants d'une expression mathématique. Chaque règle de la grammaire est associée à une relation spatiale différente qui décrit les relations spatiales entre ses constituants. Les relations sont associées à des fonctions de coût qui pénalisent plus ou moins les hypothèses selon la position et la taille relatives de leurs constituants par rapport à une référence. Ces positionnements idéaux sont très difficiles à définir, à cause de la nature floue des relations spatiales entre les symboles. En conséquence, nous essayons d'apprendre ces relations au lieu de définir les positions des symboles dans chaque relation par une règle empirique.

3.2. Le protocole d'apprentissage

L'apprentissage du système se déroule en deux phases distinctes. D'abord, le système est entraîné sur la base d'apprentissage d'expressions sans tenir compte de la grammaire. L'objectif de cette phase est l'apprentissage du classifieur de symboles. En même temps le classifieur apprend ce que nous appelons la classe de rejet. Cette classe permet d'identifier les mauvaises hypothèses et en conséquence de pénaliser ces hypothèses.

Deuxièmement, et sans tenir compte du classifieur, le système apprend les relations spatiales. En forçant la bonne reconnaissance des expressions, nous construisons les histogrammes des différences de positions et de tailles de symboles intervenant dans la relation. Pour chaque relation, deux histogrammes sont construits pour chaque élément fils de cette relation. Le premier histogramme reflète la

pertinence de taille de cet élément par rapport à la sous-expression produite par la règle reliée à cette relation. Le deuxième histogramme reflète l'alignement des élément par rapport à la sous-expression produite. Donc, cette étape construit dans un premier temps N histogrammes pour toutes les relations. Par exemple, la relation « super script » est associée à 4 histogrammes, deux pour les différences de taille (dh) pour chaque fils, et deux pour les différences d'alignement (dy). Par exemple, dh_x est calculé par la formule :

$$dh_x = \frac{h_r - h_x}{h_r} \quad [1]$$

Où les informations spatiales de la sous-expression produite (h_r, y_r) sont calculées en fonction d'informations de ces fils selon le type de relation, comme par exemple dans la Figure 5 : x et 2 sont les fils de la sous-expression x^2 via la relation « superscript ».

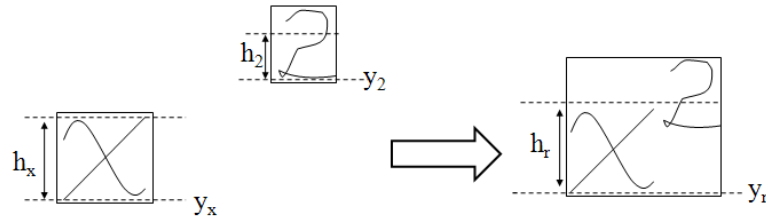


Figure 5 Illustration d'informations spatiales pour la relation « superscript »

Les différences dh, dy sont donc les écarts normalisés de position et taille de chaque nœud fils par rapport à la sous-expression. Les histogrammes résultants représentent donc les occurrences des dh et dy pour chaque élément d'une relation.

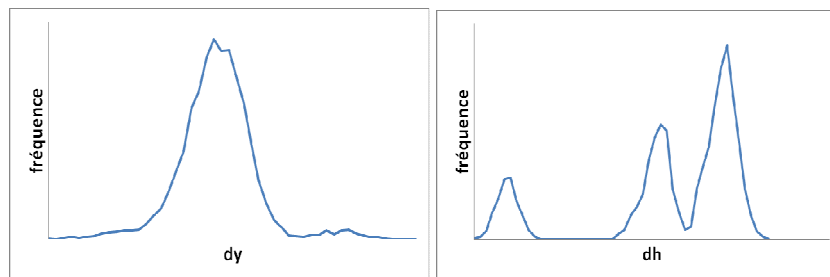


Figure 6 Exemple de 2 histogrammes de la relation "super script"

Ces histogrammes modélisant les relations spatiales, voir Figure 6, sont utilisés directement pour déduire les coûts structurels des hypothèses.

3.3. Combinaison de coûts

Le coût de reconnaissance d'une sous-expression est donné par :

$$coût_{sous-exp} = coût_{reco} + \alpha.coût_{struct} \quad [2]$$

Où le facteur alpha sert de pondération entre le coût de reconnaissance et le coût structurel. Puisque, les scores de reconnaissance et les scores structurels sont en forme de probabilité (p), nous utilisons une fonction logarithmique pour transformer ces scores en coût :

$$coût = -\log(p) \quad [3]$$

Le coût final de la reconnaissance d'une expression est donc la somme des coûts des sous-expressions formant cette expression.

En ce que concerne les coûts structurels, prenons l'exemple d'un couple de sous-expressions SE1, SE2 intervenant dans la sous expression SE et relié par la relation R. La probabilité (prob) que SE1 et SE2 produisent l'expression SE est :

$$prob(SE_{dh,dy}|R) = prob(dh_{SE1}, dy_{SE1}) \times prob(dh_{SE2}, dy_{SE2}) \quad [4]$$

Au vu du grand nombre de fréquences dans les histogrammes, les probabilités déduites sont faibles. La conséquence de cette observation est que les coûts structurels sont beaucoup plus grands que les coûts de reconnaissance. C'est pourquoi nous utilisons le facteur α dans l'équation 2. Par ailleurs, nous introduisons également l'utilisation de la possibilité (possib) au lieu de la probabilité où la fréquence maximale dans l'histogramme définit l'événement certain ayant un score égal à 1, donc :

$$possib(SE_{dh,dy}|R) = possib(dh_{SE1}, dy_{SE1}) \times possib(dh_{SE2}, dy_{SE2}) \quad [5]$$

Mais ce que l'on cherche à modéliser est la relation d'une sous-expression sachant ses histogrammes $prob(R|SE_{dh,dy})$, qui est par définition de la règle Bayes:

$$prob(R|SE_{dh,dy}) = \frac{prob(SE_{dh,dy}|R).prob(R)}{prob(SE_{dh,dy})} \quad \text{mais nous pouvons ignorer}$$

$prob(SE_{dh,dy})$, car ce terme est constant :

$$prob(R|SE_{dh,dy}) \propto prob(SE_{dh,dy}|R).prob(R) \quad [6]$$

En utilisant le même raisonnement pour la possibilité :

$$possib(R|SE_{dh,dy}) \propto possib(SE_{dh,dy}|R).possib(R) \quad [7]$$

L'utilisation de la probabilité ou la possibilité de la relation en question prend en compte la distribution de toutes les relations dans la base d'apprentissage.

Dans la section suivante nous présentons quelques expérimentations et des résultats préliminaires.

4. Expérimentations

4.1. La base d'expressions

Une base de symboles mathématiques manuscrits en-lignes a été collectée auprès de 280 scripteurs. Cette base sert à générer une base d'expressions manuscrites pseudo-synthétiques. Pour accomplir cette tâche, nous avons utilisé un outil « Latex2Ink » (Awal et al., 2008) développé par notre groupe de recherche. L'objectif de cette base synthétique est de disposer d'une grande quantité d'expressions pour l'apprentissage et aussi le réglage du système. En complément, une base de vraies expressions a été collectée pour élargir les conditions d'évaluation de ce système.

Nous avons extrait du corpus « Aster » (Raman, 1994) un sous-corpus de 36 expressions différentes couvrant une majorité des domaines mathématiques. Dans ce corpus, la taille moyenne d'une expression est de 11 symboles. Le nombre total de symboles est de 415 dont 34 classes distinctes comprenant des chiffres, des lettres latines, des lettres grecques, des opérations binaires, des symboles et des fonctions élastiques. Chacune de ces 36 expressions a été générée 280 fois en utilisant les jeux de symboles des 280 scripteurs. Afin de tester le système avec une base de vraies expressions mathématiques manuscrites en-lignes, nous avons également fait saisir 2 fois chacune de ces 36 expressions en utilisant 10 scripteurs, cf. Tableau 2. Quelques exemples d'expressions de la base synthétique et la base réelle sont montrés dans la Figure 7.

	# scripteurs	# Expressions	# Symboles
Apprentissage	180	180×36 = 6480	180×412 = 74160
Test	100	100×36 = 3600	100×412 = 41200
Test (Réel)	10	2×36 = 72	2×412 = 824

Tableau 2 La constitution des bases d'expressions

$$\sum_{i=1}^n a_i + b_i = y \quad \frac{x+y^2}{k+1}$$

$$x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \frac{x^5}{5} \pm \dots = \log(1+x)$$

Expressions synthétiques

$$1 + \frac{x}{1 + \frac{x}{1 + \frac{x}{1 + \frac{x}{1 + \dots}}}}$$

Expressions réelles

$$\frac{1 + \sqrt{5}}{2} = \phi$$

Figure 7 Exemples d'expressions des bases synthétique et réelle

4.2. Evaluation du système

L'évaluation du système au niveau expressions complètes est trop globale pour être significative, et cela dans la mesure où une expression peut être très longue. En conséquence, il est important de connaître les performances aux différents niveaux intermédiaires. Nous avons donc choisi trois mesures similaires à celles utilisées récemment dans (Yamamoto et al., 2006) et (Rhee et al., 2008) afin de pouvoir comparer nos résultats. Toutefois, il est bon de rappeler que les bases d'expressions sont différentes, et donc que la validité des comparaisons des résultats qui sont présentés au Tableau 4 est limitée.

Les performances du système ont été évaluées avec ces trois mesures:

- Taux de bonne segmentation, $SegRate = \frac{\text{\#symboles bien segmentés}}{\text{\#total de symboles}}$
- Taux de bonne reconnaissance, $recoRate = \frac{\text{\#symboles bien reconnus}}{\text{\#total de symboles}}$
- Taux de bonne reconnaissance d'expressions,

$$ExpRate = \frac{\# \text{expressions bien reconnues}}{\# \text{total d'expressions}}.$$

Une expression est bien reconnue si tous ses symboles sont bien reconnus et bien interprétés. Ainsi, si l'expression « 2^x » est reconnue comme « $2x$ », bien que tous ses symboles soient bien reconnus et segmentés, l'expression est considérée mal reconnue.

4.3. Optimisation du paramètre alpha

Le choix du paramètre alpha dans la formule [2] est très important pour équilibrer l'effet des coûts structurels et de reconnaissance. Nous avons expérimentalement choisi la valeur optimale du paramètre α et la méthode de calcul des coûts structurels.

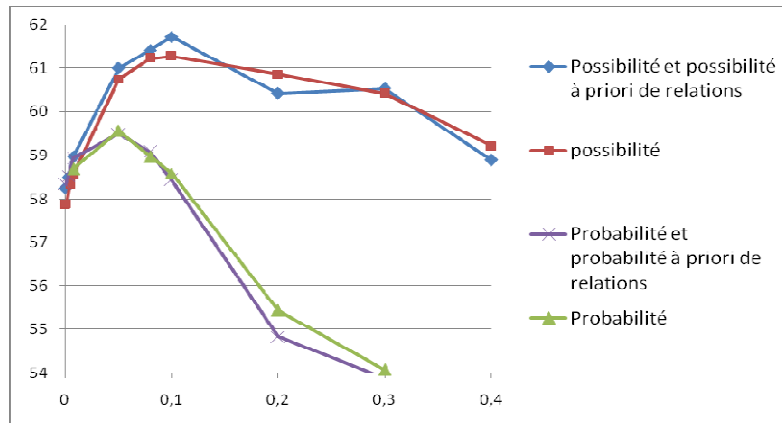


Figure 8 Evolution de taux de reconnaissance d'expressions de la base synthétique en fonction d'alpha

Sur la Figure 8, nous affichons l'évolution de taux de reconnaissance d'expressions de la base synthétique de test. En effet, l'utilisation des équations 4 et 5 pour calculer les coûts produit les courbes possibilité et probabilité. D'autre part, utiliser la probabilité (ou possibilité) à priori de la relation (eq : 6 et 7) produit les deux autres courbes. Nous constatons que l'utilisation des possibilités est moins sensible au changement de valeurs d'alpha par rapport à la probabilité. Cela s'explique du fait que la possibilité donne des scores plus forts, et en conséquence des coût plus faibles et plus adaptés pour être combinés avec les coût de reconnaissance. Les résultats présentés dans la suite sont ceux obtenus en utilisant l'équation 7, c'est-à-dire utilisant la possibilité et la possibilité à priori pour estimer le coût d'une relation associée à une règle de la grammaire.

4.4. Résultats

Dans le Tableau 3, nous comparons les résultats du système proposé dans (Awal et al., 2009) avec la méthode proposée dans cet article qui se base sur l'utilisation d'histogrammes pour modéliser les relations spatiales. Dans (Awal et al., 2009) les coûts structurels sont calculés à partir des positions et tailles des symboles par des fonctions heuristiques prédéfinies. Dans les deux cas nous utilisons un réseau de neurones à convolution (TDNN) pour la classification des symboles. Nous comparons également les résultats sur la base réelle avec les résultats du système proposé par (Rhee et al., 2008), voir Tableau 4.

Base synthétisée	SegRate %	RecoRate %	ExpRate %
(Awal 2009)	88	85.8	61.1
Méthode proposée	87.2	84.9	61.7

Tableau 3 Taux de reconnaissance sur la base synthétisée

Le Tableau 3 montre que l'apprentissage des relations spatiales améliore légèrement la performance du système sur la base de test synthétisée. Toutefois, l'utilisation des histogrammes dans la méthode proposée alourdit le système. En effet, il faut 4 histogrammes pour chaque relation avec pour chacun une quantification sur 50 à 100 valeurs, augmentant donc le nombre de paramètres libres du système. En se contentant de valeurs prédéfinies pour fixer les positions et tailles des symboles dans les relations, seuls 4 ou 6 paramètres par relation sont nécessaires. Nous pensons que les performances en utilisant les relations apprises peuvent encore évoluer si ces relations sont apprises itérativement dans une approche discriminante au lieu d'une modélisation descriptive.

Base Réelle	SegRate %	RecoRate %	ExpRate %
(Rhee 2008)	94.8	84.8	29.2
(Awal 2009)	78.5	72.3	25.7
Méthode proposée	76.9	70.4	17.1

Tableau 4 Taux de reconnaissance sur la base réelle

La bonne performance sur la base synthétisée ne se généralise pas sur la base réelle, voir Tableau 4. Cette chute de performance vient principalement de la forte dépendance avec les relations apprises sur la base d'apprentissage qui est la base synthétique et qui ne correspond pas complètement aux situations rencontrées sur la base réelle. En conséquence, nous pensons qu'il est indispensable de modéliser ces histogrammes par des fonctions de densités, des mélanges de gaussiennes par exemple. L'autre solution est d'apprendre itérativement les relations spatiales d'une façon discriminante entre les différents types de relations. Cet apprentissage itératif améliore à la fois la modélisation des relations de la base d'apprentissage, et augmente la capacité à généraliser. Pour cela, il faudrait disposer d'une grosse base d'expressions réelles afin de pouvoir correctement apprendre ces relations spatiales.

5. Conclusion et perspective

Nous avons proposé dans cet article une nouvelle méthode d'analyse structurelle et syntaxique pour un système de reconnaissance d'expressions mathématiques manuscrites en-lignes. L'objectif de la méthode proposée est l'apprentissage des relations spatiales entre les symboles dans les expressions mathématiques. Cet apprentissage participe à la fois à pénaliser les mauvais positionnements des hypothèses et aussi à déduire des coûts structurels liés à la grammaire.

Le système est entraîné et testé avec une grosse base d'expressions synthétiques. L'expérimentation et les résultats préliminaires montrent l'intérêt de cette méthode. Néanmoins, comme nous l'avons vu, la dépendance des relations apprises à la base d'apprentissage est un inconvénient de la méthode. Nous espérons surmonter ce défaut par un apprentissage itératif et discriminant de ces relations. De plus, nous allons proposer de modéliser les relations par des fonctions de distribution de probabilité (modèle multi-Gaussien par exemple) ce qui rendra le système plus léger et augmentera sa capacité de généralisation. Une autre piste de d'amélioration de système peut être d'adopter une approche floue correspondant à la nature floue de relations spatiales dans les expressions mathématiques manuscrites.

6. Bibliographie

- Awal A.M., Cousseau R., Viard-Gaudin C. "Convertisseur d'équations LATEX2Ink", in Colloque International Francophone sur l'Ecrit et le Document, Rouen, France, 193-194 (2008).
- Awal A.M, Mouchère H., Viard-Gaudin C., "Towards handwritten mathematical expression recognition", tenth International Conference on Document Analysis and Recognition, Barcelona, Spain, 1046-1050 (2009).
- Belaid A., Haton J-P., "A syntactic approach for handwritten mathematical formula recognition", in Transactions on Pattern Analysis and Machine Intelligence, 105-111 (1984).

Awal A-M, Mouchère H, Viard-Gaudin C

- Chan K-F., Yeung D-Y., "An efficient syntactic approach to structural analysis of on-line handwritten mathematical expressions", *Pattern Recognition* 33, 375-384 (2000).
- Dimitriadis Y.A., Coronado J.L., "Towards an ART based mathematical editor that uses online handwritten symbol recognition", *Pattern Recognition* 28(6), 807-822 (1995).
- Faure C., Wang Z.X., "Automatic perception of the structure of handwritten mathematical expressions", in *Computer Processing of Handwriting*, World scientific, Singapore, 337-361 (1990).
- Fitzgerald J.A., Geiselbrechtinger F., and Kechadi, T., "Structural Analysis of Handwritten Mathematical Expressions Through Fuzzy parsing". The IASTED International Conference on Advances in Computer Science and Technology, Puerto Vallarta, Mexico, anuary 23-25, 2006, pp 151-156, (2006).
- Garain U. and Chaudhuri B., "Recognition of Online Handwritten Mathematical Expressions", *IEEE Transactions on Systems, Man and Cybernetics, Part B*, Volume 34, Issue 6, pp 2366-2376, (2004).
- Ha J., Haralick R.M., Phillips I.T., "Understanding mathematical expressions from document images", third International Conference on Document Analysis and Recognition, 956-959 (1995).
- Lehmberg S., Winkler H-J., Lang M., "A Soft-decision approach for symbol segmentation within handwritten mathematical expressions", *Int. Conference on Acoustics, Speech, and Signal Processing*, Atlanta, USA, 3434-3437 (1996).
- Marzinkewitsch R., "Operating computer algebra systems by handprinted document", *International Symposium on Symbolic and Algebraic Computation*, 411-413 (1991).
- Nakayama, Y., "A prototype pen-input mathematical formula editor", in *World Conference on Educational Multimedia and Hypermedia*, 400-407 (1993).
- Raman, T.V., "Audio system for technical readings", Cornell University, (1994).
- Ray G., John A. Fitzgerald, and Tahar Kechadi. "A Purely Online Approach to Mathematical Expression Recognition", *International Workshop on Frontiers in Handwriting Recognition*, la baule, 255-260 (2006).
- Rhee T-H, Kim J-H: Efficient search strategy in structural analysis for handwritten mathematical expression recognition. *Pattern Recognition* 42(12): 3192-3201 (2009).
- Smithies S., Novins K., Arvo J., "A Handwriting-Based Equation Editor", the Graphics Interface, Kingston, Ontario, Canada, 84-91 (1999).
- Yamamoto R., Sako S., Nishimoto T., Sagayama S., "On-Line Recognition of Handwritten Mathematical Expressions Based on Stroke-Based Stochastic Context-Free Grammar", tenth International Workshop on Frontiers in Handwriting Recognition, La Baule, France, 249-254 (2006).
- Zanibbi R., Blostein D. and Cordy J., *Recognizing Mathematical Expressions Using Tree Transformation*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 24, Issue 11, pp 1455-1467, (2002).